Practical Encoders and Decoders for Euclidean Codes from Barnes-Wall Lattices

J. Harshan*, Emanuele Viterbo*, and Jean-Claude Belfiore†

Abstract

In this paper, we address the design of high spectral-efficiency Barnes-Wall (BW) lattice codes which are amenable to low-complexity decoding in additive white Gaussian noise (AWGN) channels. We propose a new method of constructing complex BW lattice codes from linear codes over polynomial rings, and show that the proposed construction provides an explicit method of bit-labelling complex BW lattice codes. To decode the code, we adapt the low-complexity sequential BW lattice decoder (SBWD) recently proposed by Micciancio and Nicolosi. First, we study the error performance of SBWD in decoding the infinite lattice, wherein we analyze the noise statistics in the algorithm, and propose a new upper bound on its error performance. We show that the SBWD is powerful in making correct decisions well beyond the packing radius. Subsequently, we use the SBWD to decode lattice codes through a novel noise-trimming technique. This is the first work that showcases the error performance of SBWD in decoding BW lattice codes of large block lengths.

Index Terms

Barnes-Wall lattices, lattice codes, low-complexity lattice decoders.

*The authors are with the Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, Australia-3168. Email:harshan.jagadeesh@monash.edu, emanuele.viterbo@monash.edu. †The author is with the Department of Communications and Electronics, Telecom ParisTech, Paris, France, Email:belfiore@enst.fr. Parts of this work are in the proceedings of IEEE International Symposium on Information Theory (ISIT) 2012 held at Cambridge, MA, USA, and International Symposium on Mathematical Theory of Networks and Systems (MTNS) 2012 held at Melbourne, Australia.

I. Introduction

Ever since random coding schemes were demonstrated to approach the capacity of additive white Gaussian noise (AWGN) channels [1], enormous research has taken place to find *structured* coding schemes which can accomplish the same job. The need for structured coding schemes is to facilitate simpler analysis of the code structure and to achieve reduced complexity in encoding and decoding. A well known method of obtaining structured codes is to carve out a finite set of lattice points from dense lattices [2]-[5]. Such codes are referred to as lattice codes, and are usually obtained as a set of coset representatives of a suitable quotient lattice. Further, the lattice codes have the advantage of inheriting most of the code properties from the parent lattice, and as a result, the choice of the lattice is crucial to the performance of the code.

A. Motivation and contributions

In this paper, we are interested in carving lattice codes from Barnes-Wall (BW) lattices [6], [7]. Our goal is to construct efficient BW lattice codes of large block lengths which work with low-complexity encoders and decoders. In particular, efficient low-complexity decoders for complex BW lattices are readily available in [13], [15]. Therefore, if lattice codes from complex BW lattices are employed for communication over AWGN channels, then the decoders of [13], [15] can be used to recover information with low computational complexity.

In [13], the authors have proposed two low-complexity implementations of the bounded distance decoder for BW lattices, namely (*i*) the sequential bounded distance decoder, and (*ii*) the parallel bounded distance decoder. Inspired by the parallel decoder in [13], list decoders for BW lattices have been recently proposed in [15]. We note that the parallel decoders of [13] and [15] have low-complexity only when implemented on sufficiently large number of parallel processors. If the above decoders are implemented on a single processor, then the complexity advantages are lost, and specifically, the complexity of the list decoder grows larger than that of the sequential decoder in [13]. Since we are interested in lattice codes of large block lengths, we focus on the

sequential bounded distance decoder which seems more suitable for implementation (see Section V-B for more details on the complexity advantages of sequential bounded distance decoder over the list decoder). The sequential decoder in [13] was proven to correct any error up to the packing radius. However, the possibility of correct decoding is not known when the received vector falls outside the bounded decoding ball of packing radius. In a nutshell, the exact error performance of the decoder is not known. The existence of this low-complexity decoder has motivated us to study its error performance, and use it to decode BW lattice codes. We refer to this decoder as the sequential BW lattice decoder (SBWD). The contribution of this paper on the construction and decoding of complex BW lattices are given below.

1) We introduce Construction A' of lattices which enables us to generate some well structured lattices from linear codes over *polynomial rings* [24]. As an immediate application, we apply Construction A' to obtain BW lattices of dimension 2^m for any $m \geq 1$. The proposed method is yet another construction of BW lattices (shown in Section III) and shows a new connection between codes over polynomial rings and lattices. We show that the proposed construction provides an explicit method of obtaining and bit-labelling complex BW lattice codes.
2) We study the error performance of the SBWD in AWGN channels. Since the error performance of the SBWD depends on the error performance of the underlying soft-input Reed-Muller (RM) decoders, we study the error performance of the soft-input RM decoder as used in the SBWD. First, we use the Jacobi-Theta functions [26] to characterize the virtual binary channels that arise in the decoding process. Subsequently, we study the noise statistics in the algorithm, and provide an upper bound on the error performance of the soft-input RM decoders. Through computer simulations, we obtain the error performance of the SBWD, and show that the decoder is powerful in making correct decisions well beyond the packing radius [25] (see Table I in Section V for the effective radius of the

SBWD decoder). This is the first work that showcases the error performance of SBWD in decoding BW lattices of large block lengths.

3) To decode the lattice code in AWGN channels, we employ the SBWD along with a noise trimming technique, wherein the components of the received vector are appropriately scaled before passing it to the SBWD. With the noise-trimming technique, the SBWD is forced to decode to a codeword in the code which in turn improves the error performance. We refer to this decoder as the *BW lattice code decoder* (BWCD). We obtain the bit error rate (BER) of the BWCD for codes in complex dimension $4, 16$, and 64, and show that the BWCD outperforms the SBWD by 0.5 dB.

B. Prior work on Barnes-Wall lattices

The BW lattices [6] is a special family of N-dimensional lattices that exist when N is a power of 2. These lattices were originally discovered as a solution to finding extreme quadratic forms in 1959 [6]. Only in 1983, the now well known connection between BW lattices and Reed-Muller codes was discovered by [8]. This connection is found in several works [9], [10], [14] in different forms. Other than its construction through Reed Muller codes, the generator matrices of the BW lattices are also known to be obtained though Kronecker products [11], [13].

In 1989, G.D. Forney proposed a low-complexity bounded distance decoding algorithm for Leech lattices [18]. As a generalization, in the same paper, a similar algorithm has been shown to be applicable in decoding all Construction D lattices. As BW lattices are known to be obtained through Construction D [9], bounded distance decoders for BW lattices were known in principle since [18]. Only in the 1990's, explicit bounded distance decoders for BW lattices were implemented for dimension up to 32, and numerical results on the error performance were reported [19], [20]. In 2008, Micciancio and Nicolosi [13] have proposed two low-complexity implementations of bounded distance decoder for BW lattices, namely (*i*) the sequential bounded distance decoder, and (*ii*) the parallel bounded distance decoder. If $N = 2^m$ denotes the dimension

of complex BW lattice, the worst-case complexity of the decoders has been shown to be $O\left(N\log^2(N)\right)$ and $O\left(\log^2(N)\right)$ for the fully sequential decoder and the fully parallel decoder, respectively. For the fully sequential decoder, the algorithm is assumed to be implemented on a single processor, whereas for the fully parallel decoder, the algorithm is assumed to be implemented on N^2 parallel processors. Inspired by the fully parallel implementation in [13], list decoders for BW lattices have been recently proposed in [15] where the list decoder outputs a list of BW lattice points within any given radius from the target vector. The complexity of the list decoder is shown to be polynomial in the dimension of the lattice, and polynomial in the list size, which is a function of the Euclidean radius. Note that the SBWD exploits the Construction D structure of BW latices as a multilevel code of nested binary Reed-Muller (RM) codes, and decodes each RM code through a successive interference cancellation technique. On the other hand, the list decoder does not exploit construction D structure of BW lattices, and hence, does not need the support of any soft-input RM decoders.

The rest of this paper is organized as follows: In Section II, we provide a background on lattice constructions from linear codes. In Section III, we introduce Construction A' of complex BW lattices. In Section IV, we study the error performance of the SBWD, while in Section V and Section VI, we use the SBWD to decode the BW lattice code. Finally, in Section VII, we conclude this paper and provide some directions for future work.

***Notations**:* Throughout the paper, boldface letters and capital boldface letters are used to represent vectors and matrices, respectively. For a complex matrix $\mathbf{X}$, the matrices $\mathbf{X}^T$, $\Re(\mathbf{X})$ and $\Im(\mathbf{X})$ denote the transpose, real part and imaginary part of $\mathbf{X}$, respectively. The set of integers, real numbers, and complex numbers are denoted by $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{C}$, respectively. We use i to represent $\sqrt{-1}$. For an n-length vector $\mathbf{x}$, we use x_j to represent the j-th component of $\mathbf{x}$. Cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. Magnitude of a complex number x is denoted by $|x|$. The number of ways of picking n objects out of m objects is denoted by C_n^m. The symbol $\lceil\cdot\rfloor$ denotes the nearest integer of a real number, and we set $\lceil a+0.5\rfloor = a$ for any $a \in \mathbb{Z}$. Finally,

we use $\Pr(\cdot)$ to denote the probability operator.

II. Background on Lattice Construction using Linear Codes

A complex lattice Λ over $\mathbb{Z}[i]$ is a discrete subgroup of $\mathbb{C}^n$ [9]. Alternatively, Λ is a $\mathbb{Z}[i]$-module generated by a basis set $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \mid \mathbf{v}_j \in \mathbb{C}^n\}$ as $\Lambda = \left\{\sum_{j=1}^{n} q_j \mathbf{v}_j \mid \forall q_j \in \mathbb{Z}[i]\right\}$. It is well known that dense lattices can be obtained via binary linear codes [9]. Depending on the structure of the underlying linear codes, lattice construction can be categorized into different types. In this section, we recall two well known constructions for the case of complex lattices [9].

Construction A:

Definition 1: A complex lattice Λ is obtained by Construction A from the binary linear code $\mathcal{C}$ if Λ can be represented as

$$\Lambda = (1+i)\mathbb{Z}[i]^n \oplus \mathcal{L}_0, \tag{1}$$

where $\mathcal{L}_0 = \{\psi(\mathbf{c}) \mid \forall \mathbf{c} \in \mathcal{C}\} \subseteq \Lambda$ is a lattice code obtained by the component-wise mapping $\psi : \mathbb{F}_2 \to \mathbb{Z}[i]$ given by $\psi(0) = 0$ and $\psi(1) = 1$ on the alphabet of $\mathcal{C}$, where $\mathbb{F}_2 = \{0, 1\}$.

Construction D:

Definition 2: A complex lattice Λ is obtained by Construction D from a family of nested binary linear codes $\mathcal{C}_{m-1} \supseteq \mathcal{C}_{m-2} \supseteq \ldots \supseteq \mathcal{C}_1 \supseteq \mathcal{C}_0$ if Λ can be represented as

$$\Lambda = (1+i)^m\mathbb{Z}[i]^n \oplus (1+i)^{m-1}\mathcal{L}_{m-1} \oplus \cdots \oplus (1+i)\mathcal{L}_1 \oplus \mathcal{L}_0, \tag{2}$$

where $\mathcal{L}_j = \{\psi(\mathbf{c}) \mid \forall \mathbf{c} \in \mathcal{C}_j\}$ is obtained by the component-wise mapping $\psi : \mathbb{F}_2 \to \mathbb{Z}[i]$ given by $\psi(0) = 0$ and $\psi(1) = 1$ on the alphabet of $\mathcal{C}_j$.

A BW lattice can be obtained via construction D as a $\mathbb{Z}[i]$ lattice as follows [10]. Suppose we want to construct the complex lattice BW_{2^m} of dimension 2^m where $m \geq 1$, let $\mathcal{RM}(r, m)$ be

the binary Reed-Muller (RM) code (Sec. 3.7, Ch. 3, [16]) of length 2^m and of order $0 \leq r \leq m$. Then, BW_{2^m} can be constructed as

$$BW_{2^m} = \left\{ (1+i)^m \mathbf{a} + \sum_{r=0}^{m-1} (1+i)^r \psi(\mathbf{c}_r) \mid \forall \mathbf{c}_r \in \mathcal{RM}(r,m), \forall \mathbf{a} \in \mathbb{Z}[i]^{2^m} \right\} \tag{3}$$

where $\psi(\cdot)$ is as given in Definition 2. For notational convenience, we also write (3) as

$$BW_{2^m} = (1+i)^m \mathbb{Z}[i]^{2^m} \oplus \bigoplus_{r=0}^{m-1} (1+i)^r \mathcal{RM}(r,m). \tag{4}$$

This method generates BW_{2^m} as a multi-level structure of nested RM codes and hence it falls under Construction D [9].

Generalized construction A [12] [17]:

Apart from Construction D, the BW lattice codes can be obtained by the generalized construction A. For the complex BW lattice BW_{2^m}, let $\mathbf{G}_{\text{BW}} \in \mathbb{C}^{N \times N}$ denote the generator matrix in the triangular form, where the rows $\{\mathbf{g}_1, \ \mathbf{g}_2, \ldots, \mathbf{g}_N\}$ of $\mathbf{G}_{\text{BW}}$ forms a basis set of BW_{2^m}, where $N = 2^m$. Let $d_1, d_2, \ldots, d_N$ represent the diagonal elements, where $d_j = (1+i)^{m_j}$ for some integer $m_j \geq 0$, and $d = (1+i)^{\max m_j}$. For this lattice construction, one can easily map binary data to lattice points as follows:

1) **Bit Labelling:** Map $\log_2(L_j)$ information bits to $a_j \in \mathbb{Z}[i]/p_j\mathbb{Z}[i]$ where $p_j = \frac{d}{d_j}$ and L_j is the cardinality of $\mathbb{Z}[i]/p_j\mathbb{Z}[i]$.
2) **Encoding:** Using $\{a_1, a_2, \ldots, a_N\}$, a lattice point is obtained as $\sum_{j=1}^{N} a_j \mathbf{g}_j$.
3) **Shaping:** Since $\Lambda = d\mathbb{Z}[i]^N + \mathcal{L}$, a lattice point within $\mathcal{L}$ can be obtained as $\bar{\mathbf{x}} = \mathbf{x} \bmod d\mathbb{Z}[i]^N$.

Motivation for Construction A':

In the bit-labelling step above, binary digits have to be mapped to the symbols of $\mathbb{Z}[i]/p_i\mathbb{Z}[i]$. Some of the well-known bit-labelling methods include gray-mapping and set-partitioning based

methods.[1] Unlike the case of real integer lattice, $\mathbb{Z}[i]/p_i\mathbb{Z}[i]$ is an arbitrary subset of $\mathbb{Z}[i]$, and bit mapping to $\mathbb{Z}[i]/p_j\mathbb{Z}[i]$ is not straightforward unless the set of representatives for $\mathbb{Z}[i]/p_j\mathbb{Z}[i]$ is chosen with good shaping property. Through Construction A', we facilitate bit-labelling on complex integers by using the truncated binary expansion of the elements of $\mathbb{Z}[i]/p_j\mathbb{Z}[i]$ over the base $1+i$ [23]. With this, the bits labelled on $a_j \in \mathbb{Z}[i]/p_j\mathbb{Z}[i]$ are nothing but the bits in the truncated binary expansion of a_j. To assist the bit-labelling step, we use polynomial rings over $\mathbb{F}_2$ to represent the elements of $\mathbb{Z}[i]/p_j\mathbb{Z}[i]$. For the encoding step, we use a linear code over polynomial rings, and obtain the lattice points as embedding of the codewords a of linear code into the Euclidean space. Finally, for the shaping step, we propose an appropriate mapping on $\mathbb{Z}[i]$ which provides a label code with appropriate shaping property, i.e., we explicitly provide a method of bit-labelling complex BW lattices. Our construction is an extension of Construction A and hence we refer to it as Construction A'. We now define polynomial rings and codes over polynomial rings.

Definition 3: (Ch. 4 in [16]) We define the polynomial quotient ring $\mathcal{U}_m = \mathbb{F}_2[u]/u^m$ in variable u for any $m \geq 1$ as

$$\mathcal{U}_m = \left\{ \sum_{k=0}^{m-1} b_k u^k \bmod u^m \mid b_k \in \mathbb{F}_2 \right\},$$

with regular polynomial addition and multiplication over $\mathbb{F}_2$ coefficients along with the quotient operation $u^m = 0$, which is equivalent to cancelling all the terms of degree greater than or equal to m.

Definition 4: A linear code $\mathcal{C}$ over $\mathcal{U}_m$ is a subset of $\mathcal{U}_m^n$ which can be obtained through a generator matrix $\mathbf{G} \in \mathcal{U}_m^{k\times n}$ as

$$\mathcal{C} = \{\mathbf{zG} \mid \forall \mathbf{z} \in \mathcal{U}_m^k\},$$

[1]Unlike uncoded communication, gray-mapping on $\mathbb{Z}[i]/p_i\mathbb{Z}[i]$ is not necessarily optimal since it does not guarantee that the neighbouring lattice points in the lattice code are separated by maximum number of information bits. Efficient bit labelling of lattice codes is a separate problem of its own and is out of the scope of this work.

for some $k \leq n$ and the matrix multiplication is over the ring $\mathcal{U}_m$.

III. Construction A' of BW Lattice

We now introduce Construction A' in the following definition.

Definition 5: A complex lattice Λ is obtained by Construction A' from a linear code $\mathcal{C}$ over $\mathcal{U}_m$ for some $m \geq 1$ if Λ can be written as

$$\Lambda = \Phi(u^m)\mathbb{Z}[i]^n + \mathcal{EC}, \tag{5}$$

where $\mathcal{EC} = \{\Phi(\mathbf{c}) \mid \forall \mathbf{c} \in \mathcal{C}\} \subseteq \mathbb{Z}[i]^n$ is a lattice code obtained from the linear code $\mathcal{C}$ through the mapping $\Phi : \mathcal{U}_m \to \mathbb{Z}[i]$ given by

$$\Phi\left(\sum_{j=0}^{m-1} b_j u^j\right) = \sum_{j=0}^{m-1} \psi(b_j)\left(\Phi(u)\right)^j,$$

such that $\psi : \mathbb{F}_2 \to \mathbb{Z}[i]$ given by $\psi(0) = 0$ and $\psi(1) = 1$, and $\Phi(u) = 1 + i$.

Note that Construction A can be obtained as a special case from Construction A' when $m = 1$, wherein the embedding operation Φ coincides with ψ given in Definition 2. In the following subsections, we use Construction A' to obtain complex BW lattices of dimension 2^m for any $m \geq 1$ by embedding a linear code $\mathcal{C}$ (denoted by $\mathcal{C}_{2^m}$) over the quotient ring $\mathcal{U}_m$ to a lattice code $\mathcal{EC}$ (denoted by $\mathcal{EC}_{2^m}$).

A. Linear codes for construction A'

In order to obtain BW_{2^m} as Construction A', we first need to find a suitable linear code $\mathcal{C}_{2^m}$ over the ring $\mathcal{U}_m$. We propose such a linear code which can be obtained by the following the generator matrix

$$\mathbf{G}_{2^m} = \begin{bmatrix} 1 & 1 \\ 0 & u \end{bmatrix}^{\otimes m},$$

where the tensor operation is over the ring $\mathcal{U}_m$.

Example 1: To obtain BW_4, the linear code $\mathcal{C}_4$ can be generated using

$$\mathbf{G}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & u & 0 & u \\ 0 & 0 & u & u \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathcal{U}_2^{4\times 4}.$$

Encoding of linear code $\mathcal{C}_{2^m}$

By using $\mathbf{G}_{2^m}$ as a matrix over $\mathcal{U}_m$, the code $\mathcal{C}_{2^m}$ is obtained as follows: Let $\mathbf{z} \in \mathcal{U}_m^{2^m}$, i.e., the j-th component of $\mathbf{z}$ is given by

$$z_j = \sum_{k=0}^{m-1} b_{k,j} u^k, \tag{6}$$

where $b_{k,j} \in \mathbb{F}_2$ for all k, j. Using $\mathbf{z}$ and $\mathbf{G}_{2^m}$, the code $\mathcal{C}_{2^m} \subseteq \mathcal{U}_m^{2^m}$ can be obtained as

$$\mathcal{C}_{2^m} = \left\{ \mathbf{x} = \mathbf{z}\mathbf{G}_{2^m} \mid \forall \mathbf{z} \in \mathcal{U}_m^{2^m} \right\}, \tag{7}$$

where the matrix multiplication is over $\mathcal{U}_m$.

We now provide an example for the proposed encoding technique, showing the positions of the information bits that get encoded to the codewords of $\mathcal{C}_{2^m}$.

Example 2: For $m = 2$, the input vector $\mathbf{z}$ and the generator matrix $\mathbf{G}_4$ are of the form,

$$\mathbf{z}^T = \begin{bmatrix} b_{0,1} + b_{1,1}u \\ b_{0,2} \\ b_{0,3} \\ 0 \end{bmatrix} \text{ and } \mathbf{G}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & u & 0 & u \\ 0 & 0 & u & u \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

We define the rate of the linear code $\mathcal{C}_{2^m}$ as the ratio of the number of information bits per codeword and the length of the code (which is also known as the spectral-efficiency of the code).

Proposition 1: The rate of the code $\mathcal{C}_{2^m}$ is $\frac{m}{2}$.

Proof: Each component of $\mathbf{z}$ carries m information bits in the variables $b_{k,j}$ as shown in (6). This amounts to a total of $m2^m$ bits carried by $\mathbf{z}$. However, since the matrix multiplication is over $\mathcal{U}_m$, not all the information bits $b_{k,j}$ are encoded as codewords of $\mathcal{C}_{2^m}$ (since $u^k = 0$ for $k \geq m$). Using the structure of $\mathbf{G}_{2^m}$ it is possible to identify the indices (k, j) of information bits $b_{k,j}$ which get encoded into the codewords of $\mathcal{C}_{2^m}$ as follows. Let the set $\mathcal{I}_q$ denote the indices of the rows of $\mathbf{G}_{2^m}$ whose components take values 0 or u^q for each $q = 0, 1, \ldots, m-1$. Due to the quotient operation $u^m = 0$, the components of $\mathbf{z}$ which are in the index set $\mathcal{I}_q$ are restricted to be of the form,

$$z_j = \sum_{k=0}^{m-1-q} b_{k,j} u^k \ \forall j \in \mathcal{I}_q.$$

For example, $z_1 = \sum_{k=0}^{m-1} b_{k,1} u^k$ and $z_{2^m} = 0$. Using the structure of $\mathbf{G}_{2^m}$ we observe that the cardinality of $\mathcal{I}_q$ denoted by $|\mathcal{I}_q|$ is C_q^m, and hence we find the total number of information bits per codeword of $\mathcal{C}_{2^m}$ as $\sum_{k=0}^{m-1}(m-k)C_k^m = \frac{m}{2}2^m$. ■

We now show the equivalence of our encoding technique to Construction D. In other words, the following theorem shows that the codewords generated in (7) can be uniquely represented as vectors of a multi-level code of nested RM codes.

Theorem 1: The codewords generated in (7) can be uniquely represented as codewords obtained through Construction D.

Proof: See the proof of Theorem 1 in [24]. ■

Till now, we have presented the linear code $\mathcal{C}_{2^m}$ and its encoding technique over the quotient ring $\mathcal{U}_m$. Now, we discuss the embedding operation of $\mathcal{C}_{2^m}$ into the Euclidean space. By using the map $\Phi(u) = 1 + i$ on $\mathcal{C}_{2^m}$, we get the lattice code $\mathcal{EC}_{2^m}$. Note that $\mathcal{EC}_{2^m}$ can be used as a *tile* in constructing the BW lattice, i.e., BW_{2^m} can be obtained by replicating $\mathcal{EC}_{2^m}$ in $\mathbb{Z}[i]^{2^m}$ as $BW_{2^m} = (1+i)^m \mathbb{Z}[i]^{2^m} + \mathcal{EC}_{2^m}$. It can be verified that $\mathcal{EC}_{2^m}$ is an arbitrary subset of BW_{2^m} and does not have cubic shaping. In Fig. 1, we plot the complex points generated as

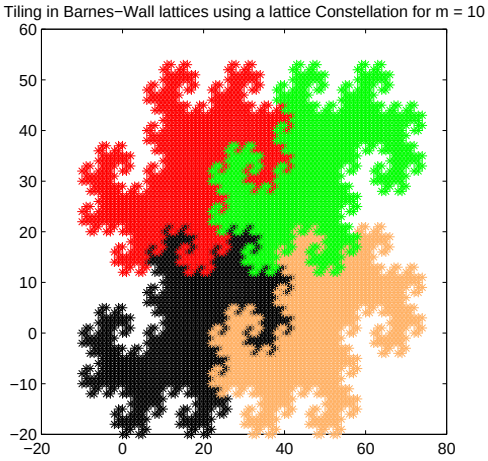

Fig. 1. Filling the complex plane using the tile generated by $\sum_{r=0}^{m-1}(1+i)^r b_r$ for $m = 10$.

$\left\{\sum_{r=0}^{m-1}(1+i)^r b_r \mid b_r \in \{0,1\}\right\}$ for $m = 10$. Note that the points generated by $\sum_{r=0}^{m-1}(1+i)^r b_r$ are marked in black, whereas the points in other shades correspond to the shifted version of $\sum_{r=0}^{m-1}(1+i)^r b_r$ by constants $(1+i)^m, i(1+i)^m$ and $(1+i)(1+i)^m$.

Note that the code $\mathcal{EC}_{2^m}$ does not have good shaping, we observe that the average transmit power of the scheme is not small. To fix this problem, we propose a one-to-one mapping ϕ on $\mathcal{EC}_{2^m}$ to obtain a new lattice code denoted by $\mathcal{L}_{2^m}$ such that it has good shaping property.

B. BW lattice codes with cubic shaping

Here, we propose a one-to-one mapping ϕ on $\mathcal{EC}_{2^m}$ to obtain a new lattice code $\mathcal{L}_{2^m}$ which has the cubic shaping property when m is even, and the rectangular shaping property when m is odd. For any $\mathbf{x} = [x_1, x_2, x_3, \ldots, x_{2^m}] \in \mathcal{EC}_{2^m}$, the mapping ϕ operates on each component of $\mathbf{x}$ as,

$$\phi(x_j) = \begin{cases} x_j \bmod 2^{\frac{m}{2}}, & \text{when } m \text{ is even;} \\ \varphi\left(x_j \bmod 2^{\frac{m+1}{2}}\right), & \text{when } m \text{ is odd,} \end{cases} \tag{8}$$

where $\varphi(\cdot)$ is defined on $\mathbb{Z}_{2^{\frac{m+1}{2}}}[i]$ as

$$\varphi(z) = \begin{cases} z, \text{ when } \Im(z) < 2^{\frac{m-1}{2}}; \\ z + \left(2^{\frac{m-1}{2}} - i2^{\frac{m-1}{2}}\right), \text{ when } \Re(z) < 2^{\frac{m-1}{2}} \\ \qquad \text{and } \Im(z) \geq 2^{\frac{m-1}{2}}; \\ z - \left(2^{\frac{m-1}{2}} + i2^{\frac{m-1}{2}}\right), \text{ when } \Re(z) \geq 2^{\frac{m-1}{2}} \\ \qquad \text{and } \Im(z) \geq 2^{\frac{m-1}{2}}. \end{cases} \tag{9}$$

The mapping ϕ guarantees the following property on $\mathcal{L}_{2^m}$:

$$\mathcal{L}_{2^m} \subseteq \begin{cases} \left\{\mathbb{Z}_{2^{\frac{m}{2}}}[i]\right\}^{2^m}, \text{ if } m \text{ is even;} \\ \\ \left\{\mathbb{Z}_{2^{\frac{m+1}{2}}}\right\}^{2^m} + i\left\{\mathbb{Z}_{2^{\frac{m-1}{2}}}\right\}^{2^m}, \text{ if } m \text{ is odd.} \end{cases} \tag{10}$$

From (10), note that each component of the vector in $\mathcal{L}_{2^m}$ is in a cubic box and a rectangular box, when m is even and odd, respectively. In Fig. 2, we present the complex points $\sum_{r=0}^{m-1}(1+i)^r b_r$ with and without the mapping ϕ for $m = 10$. With this, the lattice code $\mathcal{L}_{2^m}$ can be obtained from $\mathcal{C}_{2^m}$ through the composition map

$$\chi = \phi(\Phi(\cdot)), \tag{11}$$

where Φ and ϕ are given in Definition 5 and (8) respectively. The following proposition shows that $\chi(\cdot)$ is a one-to-one map on $\mathcal{C}_{2^m}$

Proposition 2: The mapping χ given in (11) is one-to-one.

Proof: Since χ is a composition mapping of Φ and ϕ, and $\Phi(\cdot)$ is a substitution operation using binary representation of complex numbers over the base $(1+i)$, we have to prove that ϕ given in (8) is one-to-one. Here, we provide the proof when m is even. For any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{EC}_{2^m}$ such that $\mathbf{x}_1 \neq \mathbf{x}_2$, we prove that $\phi(\mathbf{x}_1) \neq \phi(\mathbf{x}_2)$. Applying the modulo operation in (8), $\mathbf{x}_j$ satisfies $\mathbf{x}_j = 2^{\frac{m}{2}}\mathbf{r}_j + \phi(\mathbf{x}_j)$ for each $j = 1, 2$, where $\phi(\mathbf{x}_j) \in \mathcal{L}_{2^m}$ and $\mathbf{r}_j \in \mathbb{Z}[i]^{2^m}$. This implies

$$\phi(\mathbf{x}_j) = \mathbf{x}_j - 2^{\frac{m}{2}}\mathbf{r}_j = \mathbf{x}_j + (1+i)^m \mathbf{r}'_j, \tag{12}$$

for some $\mathbf{r}'_j \in \mathbb{Z}[i]^{2^m}$. The second equality follows as $(1+i)^m = a2^{\frac{m}{2}}$, where $a \in \{1, -1, i, -i\}$. Further, since each component of $\mathbf{x}_j$ is of the form $\sum_{r=0}^{m-1}(1+i)^r b_r$ for $b_r \in \{0, 1\}$, the R.H.S

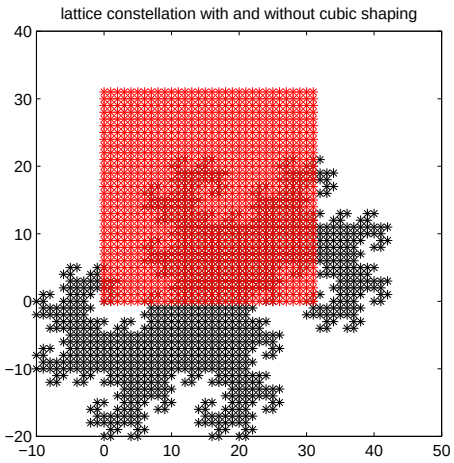

Fig. 2. Complex points generated by $\sum_{r=0}^{m-1}(1+i)^r b_r$ and $\phi(\sum_{r=0}^{m-1}(1+i)^r b_r)$ for $m = 10$.

of (12) is nothing but the binary decomposition of $\phi(\mathbf{x}_j)$ over the base $(1+i)$. Since the radix representation over $(1+i)$ is unique, we have $\phi(\mathbf{x}_1) = \phi(\mathbf{x}_2)$ only if $\mathbf{x}_1 = \mathbf{x}_2$. This completes the proof when m is even. The one-to-one nature of ϕ can be proved on the similar lines when m is odd. ■

The above proposition implies that mapping ϕ provides a new lattice code with better shaping property. The following theorem shows that $\mathcal{L}_{2^m}$ can be used as a tile to obtain BW lattices.

Theorem 2: The lattice code $\mathcal{L}_{2^m}$ and the lattice BW_{2^m} are related as $BW_{2^m} = (1+i)^m\mathbb{Z}[i]^{2^m} \oplus \mathcal{L}_{2^m}$.

Proof: See the proof of Theorem 2 in [24]. ■

Using the results of Theorem 2, BW_{2^m} is given by $BW_{2^m} = (1+i)^m\mathbb{Z}[i]^{2^m} \oplus \mathcal{L}_{2^m}$, where $\mathcal{L}_{2^m}$ is the lattice code obtained from $\mathcal{C}_{2^m}$ through the mapping $\chi = \phi(\Phi(\cdot))$ on $\mathcal{U}_m$.

IV. On the Error Performance of the SBWD

In this section, we study the error performance of the SBWD in decoding the infinite BW lattice. In [13], it is shown that for $\mathbf{x} \in BW_{2^m}$, if there exists $\mathbf{y} \in \mathbb{C}^{2^m}$ such that $d^2_{min}(\mathbf{x}, \mathbf{y}) \leq \frac{N}{4}$,

where $N = 2^m$, then the SBWD correctly finds (or decodes) the lattice point $\hat{\mathbf{x}} = \mathbf{x}$. In the context of using SBWD in AWGN channels, the vector $\mathbf{y}$ corresponds to $\mathbf{y} = \mathbf{x} + \mathbf{n}$, where $\mathbf{x} \in BW_{2^m}$ and $n_j \sim \mathcal{CN}(0, \sigma^2)\ \forall j$. This implies that the codeword error rate (CER) of the SBWD given by $\Pr(\hat{\mathbf{x}} \neq \mathbf{x})$ is upper bounded as

$$\Pr(\hat{\mathbf{x}} \neq \mathbf{x}) \leq \Pr\left(|\mathbf{n}|^2 > \frac{N}{4}\right).$$

Note that $\frac{\sqrt{N}}{2}$ is the packing radius of BW_{2^m}, and hence the above bound is the well known *sphere upper bound* (SUB) [22]. In [13], the focus was only on the complexity of the decoder but not on the analysis of the tightness of the SUB. In other words, the possibility of correct decision is not known when $|\mathbf{n}|^2 > \frac{N}{4}$. We study the error performance and show that the decoder is powerful in making correct decisions well beyond the packing radius. Without loss of generality, we study the error performance when the zero lattice point is transmitted. We analyze the SBWD algorithm and point out the reason for the improvement in the error performance (with reference to the SUB). We first recall the SBWD algorithm of [13].

The Sequential BW Lattice Decoding Algorithm:

function SEQBW(r, $\mathbf{y}$)

if $\mathbf{y} \in \mathbb{C}^N$ and $N \leq 2^r$

return $\lceil \mathbf{y} \rfloor$;

else

$\mathbf{b} = \lceil \Re(\mathbf{y}) \rfloor + \lceil \Im(\mathbf{y}) \rfloor \bmod 2$;

$\rho = 1 - 2(\max(|\lceil \Re(\mathbf{y}) \rfloor - \Re(\mathbf{y})|, |\lceil \Im(\mathbf{y}) \rfloor| - \Im(\mathbf{y})))$;

$\hat{\mathbf{c}} = \text{RMDEC}(r, \mathbf{b}, \rho)$;

$\mathbf{v} = \text{SEQBW}(r+1, (\mathbf{y} - \hat{\mathbf{c}})/(1+i))$;

return $\hat{\mathbf{c}} + (1+i)\mathbf{v}$;

end if

end function

The above decoder is a successive interference cancellation (SIC) type decoder which exploits the BW lattice structure as a multi-level code of nested RM codes (as per Construction D). At each level, the algorithm uses a variant of the soft-input RM decoder [21] (denoted by the function RMDEC which is given as Algorithm 3 in [13]) to decode, and cancel the RM codeword at that level. Therefore, the error performance of the SBWD is fundamentally determined by the error performance of the underlying soft-input RM decoders. In particular, we have

$$\Pr(\hat{\mathbf{x}} \neq \mathbf{x}) = \Pr\left(\bigcup_{r} \mathcal{E}(\hat{\mathbf{c}}_r \neq \mathbf{c}_r)\right), \tag{13}$$

where $\mathcal{E}(\hat{\mathbf{c}_r} \neq \mathbf{c}_r)$ denotes an error event while decoding $\mathcal{RM}(r,m)$. Hence, it is important to compute $\Pr(\hat{\mathbf{c}}_r \neq \mathbf{c}_r)$ for each $\mathcal{RM}(r,m)$. Along that direction, it is necessary to model the effective binary channel induced for each RM code $\mathcal{RM}(r,m)$. We propose a model for such a binary channel which is accurate for $r=0$, while for $r \neq 0$, it neglects the error propagation in the SIC decoder algorithm. To decode the RM code at each level, a hard-decision binary value b_j is obtained from y_j as

$$b_j = \lceil \Re(y_j) \rfloor + \lceil \Im(y_j) \rfloor \bmod 2. \tag{14}$$

Due to the combination of the round and the modulo operation (henceforth referred to as the round-modulo operation) in (14), the codewords of $\mathcal{RM}(r,m)$ are passed through a virtual binary channel with the cross-over probability given by,

$$P_c = \Pr(b_j = 1 \mid c_j = 0),$$

where $\mathbf{c} \in \mathcal{RM}(r,m)$. Since the zero lattice point is transmitted, $\mathbf{c}$ is the all zero codeword for each $\mathcal{RM}(r,m)$, and hence the relevant cross-over probability is $\Pr(b_j = 1 \mid c_j = 0)$. The following theorem shows that P_c can be upper bounded by a Jacobi-Theta function [26].

Theorem 3: The cross-over probability P_c induced by the round-modulo operation in (14) is

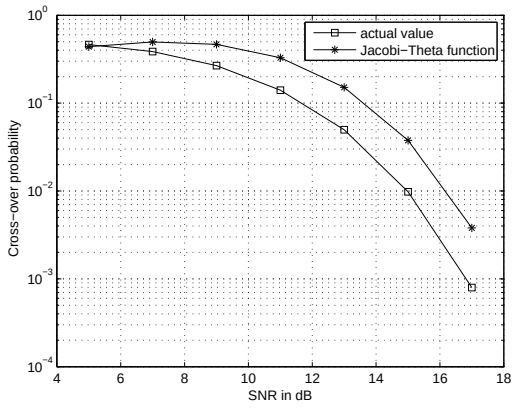

Fig. 3. Comparison of the cross-over probability with the upper bound using the Jacobi-Theta function

upper bounded as

$$P_c \leq \left(e^{-\frac{1}{4\sigma^2}}\right) \vartheta\left(\frac{i4}{\pi\sigma^2}, \frac{i}{\pi\sigma^2}\right), \tag{15}$$

where $\vartheta\left(z,\tau\right)$ is the Jacobi-Theta function given by

$$\vartheta\left(z,\tau\right) = \sum_{a=-\infty}^{\infty} e^{\pi i a^2 \tau + 2\pi i a z}.$$

Proof: We first compute P_c, and then propose an upper bound. To assist compute P_c, we compute the probability that $\Re(y_j)$ (or $\Im(y_j)$) falls within an interval $(z-0.5, z+0.5]$ centred around an integer z, when $c_j = 0$. Since the additive noise is circularly symmetric, it is sufficient to calculate the above probability for either $\Re(y_j)$ or $\Im(y_j)$. We use y to denote either $\Re(y_j)$ or $\Im(y_j)$. For the odd integer case, we have

$$\begin{aligned} P_{\text{o}} &\triangleq \sum_{a=-\infty}^{\infty} \Pr\left(2a+0.5 < y \leq 2a+1.5\right), \\ &= \sum_{a=-\infty}^{\infty} \left[\int_{2a+0.5}^{2a+1.5} \mathrm{P}_{\mathbf{y}}(y) dy\right], \\ &= \sum_{a=-\infty}^{\infty} \left[Q\left(\frac{2a+0.5}{\sigma/\sqrt{2}}\right) - Q\left(\frac{2a+1.5}{\sigma/\sqrt{2}}\right)\right], \end{aligned} \tag{16}$$

where $\mathrm{P}_{\mathbf{y}}(y)$ is the probability density function of y, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} du$, and $\sigma^2/2$ is the variance of y. For the even integer case, we have

$$\begin{aligned} P_{\mathrm{e}} &\triangleq \sum_{a=-\infty}^{\infty} \Pr\left(2a - 0.5 < y \leq 2a + 0.5\right), \\ &= \sum_{a=-\infty}^{\infty} \left[\int_{2a-0.5}^{2a+0.5} \mathrm{P}_{\mathbf{y}}(y) dy\right], \\ &= \sum_{a=-\infty}^{\infty} \left[Q\left(\frac{2a-0.5}{\sigma/\sqrt{2}}\right) - Q\left(\frac{2a+0.5}{\sigma/\sqrt{2}}\right)\right]. \end{aligned} \tag{17}$$

Note that b_j is 1 whenever $\lceil \Re(y_j) \rfloor + \lceil \Im(y_j) \rfloor$ is an odd number. This can happen when (*i*) $\lceil \Re(y_j) \rfloor$ is odd and $\lceil \Im(y_j) \rfloor$ is even, or (*ii*) $\lceil \Re(y_j) \rfloor$ is even and $\lceil \Im(y_j) \rfloor$ is odd. From (16) and (17), we can write

$$P_c = P_{\mathrm{o}}(1 - P_{\mathrm{o}}) + (1 - P_{\mathrm{o}})P_{\mathrm{o}}, \tag{18}$$

$$= 2P_{\mathrm{o}} - 2(P_{\mathrm{o}})^2. \tag{19}$$

By dropping the term $2(P_{\mathrm{o}})^2$, we upper bound P_c as

$$\begin{aligned} P_c &\leq 2P_{\mathrm{o}}, \\ &\leq 2 \sum_{a=-\infty}^{\infty} \left[Q\left(\frac{2a+0.5}{\sigma/\sqrt{2}}\right)\right], \qquad (20) \\ &\leq \sum_{a=-\infty}^{\infty} e^{-\frac{(2a+0.5)^2}{\sigma^2}}, \qquad (21) \\ &= e^{-\frac{(0.5)^2}{\sigma^2}} \sum_{a=-\infty}^{\infty} e^{\frac{-4a^2-2a}{\sigma^2}}, \\ &= \left(e^{-\frac{1}{4\sigma^2}}\right) \vartheta\left(\frac{i4}{\pi\sigma^2}, \frac{i}{\pi\sigma^2}\right), \end{aligned}$$

where the bound in (20) comes from dropping the terms of the form $Q\left(\frac{2a+1.5}{\sigma/\sqrt{2}}\right)$ in (16), and the bound in (21) is due to the Chernoff bound $Q(x) \leq \frac{1}{2} e^{\frac{-x^2}{2}}$. ■

Note that the Jacobi-Theta function can be evaluated at any pair (τ, z). In Fig. 3, the empirical values of P_c are presented along with the bound in (15) for various values of SNR $= \frac{1}{\sigma^2}$. We point out that the bound is not tight due to the Chernoff-bound on each $Q(\cdot)$ function.

It is well known that P_c determines the error-performance of a hard decision decoder. Since we have a soft-input decoder, we need to obtain the relevant statistics on the soft inputs. We now study the soft-input RM decoder used in the SBWD. Unlike the codewords of RM code in [21], the RM codewords at each level of BW lattice take values over $\{0,1\}$. The soft-input used for the RM decoder is $\rho = 1-2\mathbf{d}$, where $\mathbf{d} = \max\left(|\lceil \Re(\mathbf{y})\rfloor - \Re(\mathbf{y})|, |\lceil \Im(\mathbf{y})\rfloor - \Im(\mathbf{y})|\right)$. Also, unlike the soft metric in [21], ρ_j is bounded in the interval $[0,1]$. This is because $d_j \in [0, 0.5]$, which is a result of the round-modulo operation in (14). One could imagine $\mathbf{b}$ and ρ to be obtained from the received vector in a virtual additive noise channel, wherein each component of the received vector is always within a distance of 0.5 from either 0 or 1. Therefore, if $\mathbf{c}$ denotes a RM codeword at a particular level of the transmitted BW lattice point, then the effective noise $\mathbf{n}^{eff}$ as seen by the soft-input RM decoder at that level is of the form,

$$n_j^{eff} = \begin{cases} d_j, \text{ when } b_j = c_j; \\ 1-d_j, \text{ when } b_j \neq c_j; \end{cases} \tag{22}$$

for $1 \leq j \leq N$. Note that n_j^{eff} has bounded support in the interval $[0,1]$. For an analogy with respect to the model in [21], the code alphabet $\{0,1\}$ in [13] corresponds to the code alphabet $\{-1,1\}$ in [21] and the effective noise $\mathbf{n}^{eff}$ in [13] corresponds to the AWGN in [21]. At each level of the BW lattice, the lattice code $(1+i)^r\mathcal{RM}(r,m)$ for any $0 \leq r \leq m-1$ has the minimum squared Euclidean distance of N. By using the proposition in Section IV.A of [21], the probability of incorrect decision of the soft-input RM decoder at each level of SBWD is upper bounded as shown in the proposition below.

Proposition 3: The codeword error rate $\Pr(\hat{\mathbf{c}}_r \neq \mathbf{c}_r)$ for each $\mathcal{RM}(r,m)$ is upper bounded as,

$$\Pr(\hat{\mathbf{c}}_r \neq \mathbf{c}_r) \leq \Pr\left(|\mathbf{n}^{eff}|^2 > \frac{N}{4}\right) \text{ for } r = 0, 1, \ldots, m-1. \tag{23}$$

It is important to note that the above bound is different from $\Pr(|\mathbf{n}|^2 > \frac{N}{4})$ since $\mathbf{n}$ is Gaussian distributed. We do not have closed form expression on the distribution of either n_j^{eff} or $|n_j^{eff}|^2$. In Fig. 4, we display the histogram of the realizations of n_j^{eff} for various values of σ^2, when

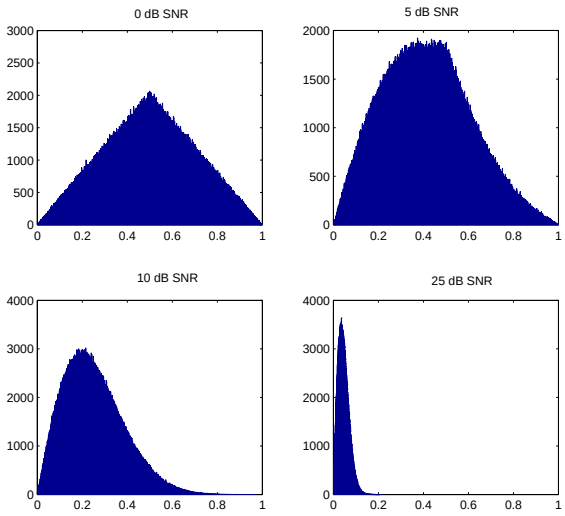

Fig. 4. Histogram of n_j^{eff} when SNR = $\frac{1}{\sigma^2}$ takes the values 0 dB, 5 dB, 10 dB and 25 dB.

the zero RM codeword is the transmitted. Note that for σ^2 = 0 dB, the histogram of n_j^{eff} has the triangular shape centred around 0.5, which implies a very high (close to 0.5) cross-over probability when obtaining the hard decision vector **b**. On the other hand, at lower values of σ^2, the distribution is skewed towards zero indicating smaller cross-over probability.

V. SBWD to Decode BW Lattice Code $\mathcal{L}_{2^m}$ for AWGN Channel

In this section, we discuss the use of SBWD to decode the lattice code $\mathcal{L}_{2^m}$. First, we describe a method to transmit the codewords of $\mathcal{L}_{2^m}$. For any $\mathbf{x} \in \mathcal{L}_{2^m}$, the transmitted vector is of the form[2]

$$\mathbf{x}_t = (2\mathbf{x} - c)\,, \tag{24}$$

[2]The transmitted vector is offset by a constant c towards the origin to reduce the average transmit energy.

where

$$c = \begin{cases} \left(2^{\frac{m}{2}} - 1\right) + i\left(2^{\frac{m}{2}} - 1\right), & \text{when } m \text{ is even;} \\ \left(2^{\frac{m+1}{2}} - 1\right) + i\left(2^{\frac{m-1}{2}} - 1\right), & \text{when } m \text{ is odd.} \end{cases} \tag{25}$$

Using the scale and the shift operation in (24), each component of $\mathbf{x}_t$ takes value from the regular 2^m-QAM constellation. In particular, the QAM constellation is square and non-square when m is even and odd, respectively. When $\mathbf{x}_t$ is transmitted, the received vector $\bar{\mathbf{y}}$ is given by

$$\bar{\mathbf{y}} = \mathbf{x}_t + \bar{\mathbf{n}}, \tag{26}$$

where $\bar{\mathbf{n}}$ is the AWGN with $\bar{n}_j \sim \mathcal{CN}(0, \sigma^2)\ \forall j$. In this section, SNR of the channel is defined as E_s/σ^2, where E_s denotes the average energy of 2^m-QAM constellation. With the inverse operation to (24) as $\mathbf{y} = \frac{1}{2}\bar{\mathbf{y}} + c$, the equivalent AWGN channel becomes

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{27}$$

where $\mathbf{x} \in \mathcal{L}_{2^m}$ and $n_j \sim \mathcal{CN}(0, \frac{\sigma^2}{4})$. We use the SBWD [13] on (27) to decode the lattice code $\mathcal{L}_{2^m}$. When a codeword of $\mathcal{L}_{2^m}$ is transmitted, the SBWD decodes to a lattice point in the infinite lattice BW_{2^m}. In such a decoding method, irrespective of whether the decoded lattice point falls in the code or not, the information bits can be recovered from the decoded RM codewords at every level of SBWD (as shown in the algorithm in Sec. IV).

A. Simulation results on the codeword error rate (CER) of SBWD

In this subsection, we present the CER of the SBWD along with some upper bounds and lower bounds. For the simulation results, we use SNR $= E_s/\sigma^2$, where E_s denotes the average energy of the regular 2^m-QAM constellation. In each of Fig. 5-9, we present (*i*) the CER of the SBWD, (*ii*) the SUB (Section IV.D, [22]), (*iii*) the sphere lower bound (SLB) (Section IV.D, [22]), (*iv*) the CER in decoding $\mathcal{RM}(0, m)$ at the first level of the SBWD, and (*v*) the upper bound on the CER in decoding $\mathcal{RM}(0, m)$ given by $\Pr(|\mathbf{n}^{eff}|^2 > \frac{N}{4})$ (obtained through simulation results by empirically generating $\mathbf{n}^{eff}$).

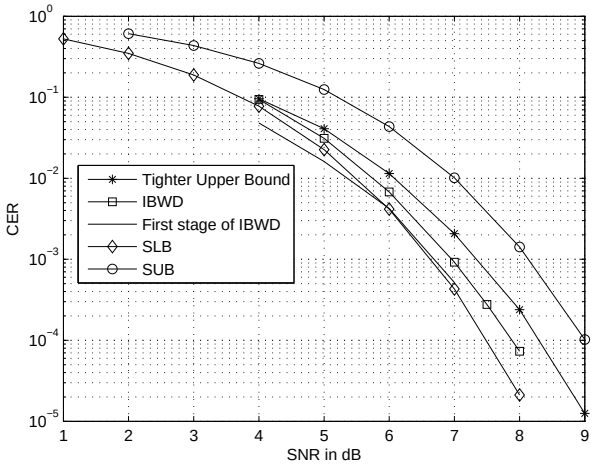

Fig. 5. CER of SBWD decoding BW_4.

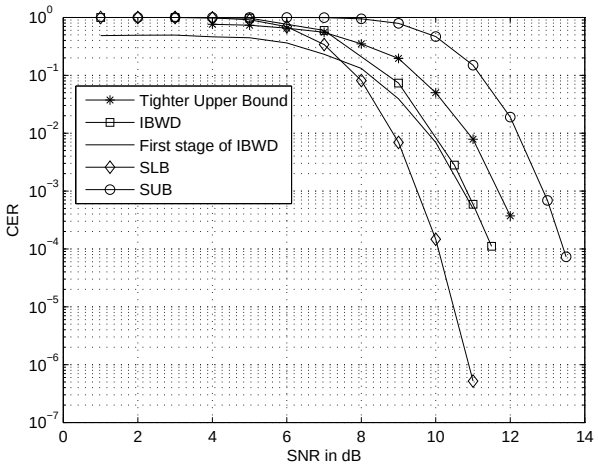

Fig. 6. CER of SBWD for decoding BW_{16}.

From Fig. 5-9, we make the following observations: the SUB is not a tight upper bound on the CER of SBWD. Also, $\Pr(|\mathbf{n}^{eff}|^2 > \frac{N}{4})$ is an upper bound on the CER of SBWD, and in particular, it is a tighter upper bound than the SUB. The CER of the soft-input RM decoder for $\mathcal{RM}(0, m)$ is a tight lower bound on the CER of the SBWD. This implies that if there is no error at the first level of the decoder, then with high probability, there will be no errors at

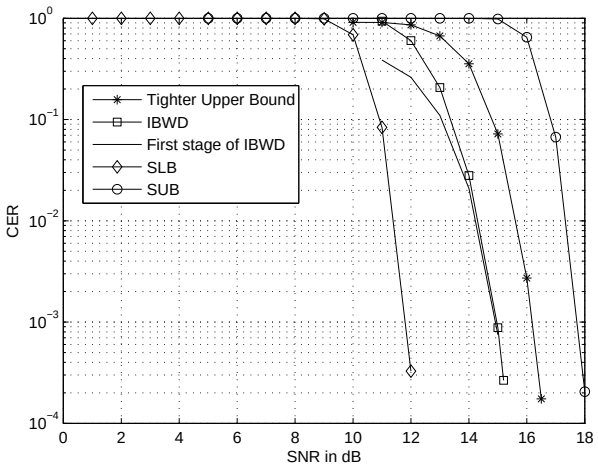

Fig. 7. CER of SBWD for decoding BW_{64}.

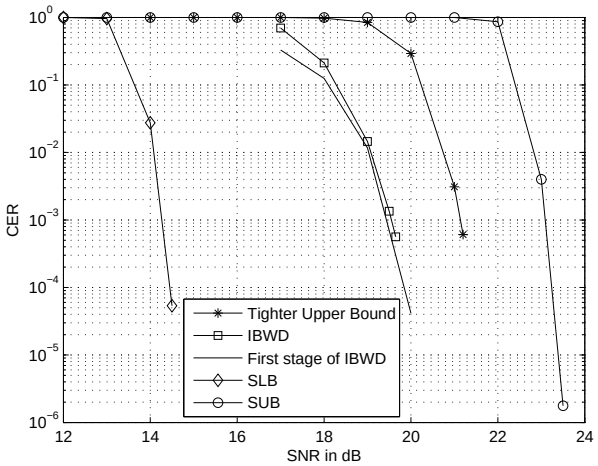

Fig. 8. CER of SBWD for decoding BW_{256}.

subsequent levels of the soft-input RM decoder. In summary, the simulation results highlight that the SBWD is quite powerful in making correct decisions even beyond the packing radius, and the deviation from the SUB increases for larger dimensions. As a result SBWD can be employed to efficiently decode lattice codes of large block lengths with low-complexity. This behaviour in the error performance of SBWD was not known in the literature.

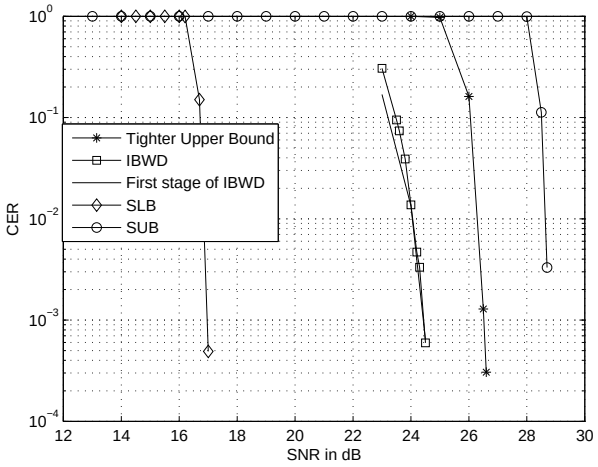

Fig. 9. CER of SBWD for decoding BW_{1024}.

B. Comparing the complexity of the SBWD with the list decoder in [15]

In this subsection, we compare the complexity of the SBWD with the BW list decoder [15]. For a fair comparison, we assume that the list decoder is implemented on a single processor. On a single processor, the complexity of the SBWD is $O(N\log^2(N))$, whereas the complexity of the list decoder is $O(N^2)(l(m,\eta))^2$, where $l(m,\eta)$ is the worst case list size at a relative squared distance of η (the relative squared distance is the squared Euclidean distance normalized by the dimension of the lattice). We compare the complexity of the two decoders for a codeword error rate of 10^{-3}. In particular, we first approximate the error performance of the SBWD as a bounded distance decoder for some radius $\bar{\eta}$, and then compute the complexity of the list decoder with the corresponding value of $\bar{\eta}$. In Table I, we display the lower bound (as given in Theorem 1.3 in [15]) on the complexity of the list decoder to achieve the error performance of SBWD. The table shows that the list decoder has higher complexity than the SBWD to achieve the same performance. In summary, for single processor implementation, SBWD can be preferred to the list decoder to decode BW lattice codes of large block lengths. However, for codeword error rates lower than that of SBWD, the list decoder has to be used, preferably on parallel processors.

TABLE I

COMPLEXITY OF THE LIST DECODER [15] TO ACHIEVE THE PERFORMANCE OF SBWD

Dimension N	$\bar{\eta}$	A lower bound on $N^2(l(m,\bar{\eta}))^2$	$N\log^2(N)$ (complexity of SBWD)
4	0.33	16	16
16	0.4	256	256
64	0.48	4096	2304
256	0.56	262144	16384
1024	0.67	1.07×10^9	102400

Table I also shows the potential of SBWD to decode well beyond the relative squared distance of $\eta = 0.25$. For complex dimensions of 256 and 1024, the effective radius of SBWD is as high as $\frac{N}{2}$ and $\frac{2N}{3}$, respectively.

VI. NOISE TRIMMING TECHNIQUE FOR THE SBWD

When a codeword of $\mathcal{L}_{2^m}$ is transmitted, the SBWD decodes to a lattice point in the infinite lattice BW_{2^m}. In such a decoding method, irrespective of whether the decoded lattice point falls in the code or not, the information bits can be recovered from the decoded RM codewords at every level of SBWD (as shown in the algorithm in Sec. IV). To further improve the error performance, we force the SBWD to specifically decode to a codeword in the lattice code, and subsequently recover the information bits, with more reliability. We refer to such a decoder as the BW lattice code decoder (BWCD). We use a technique that forces the SBWD to decode to a codeword in the lattice code $\mathcal{L}_{2^m}$. We refer to this technique as the noise trimming technique, which exploits the structure of $\mathcal{L}_{2^m}$. From (10), we know that each component of a codeword is within a rectangular box $\mathcal{B} \subseteq \mathbb{C}$. In particular, the box $\mathcal{B}$ shares its edges with $\mathbb{Z}_{2^{\frac{m}{2}}}[i]$ and $\mathbb{Z}_{2^{\frac{m+1}{2}}} + i\mathbb{Z}_{2^{\frac{m-1}{2}}}$ when m is even and odd, respectively. In order to use SBWD, and to decode to a codeword within the code, we *trim* the in-phase and quadrature components of the received

vector (the algorithm is given below) to be within a box $\mathcal{B}' \supseteq \mathcal{B}$ marginally larger than $\mathcal{B}$ by length ϵ on each dimension. Then, we feed the trimmed received vector to the SBWD and decode the information bits. Note that the choice of ϵ is crucial to decode a codeword within the code, and to improve the BER with reference to the SBWD. We now provide an algorithm for the trimming method, which works independently on the in-phase and quadrature component of the scalars in $\mathbf{y} = [y_1, y_2, \ldots, y_{2^m}]$ in (27). In particular, the algorithm presented in the sequel works on the in-phase and quadrature component of y_j when m is even. Extension to the case when m is odd is straightforward.

Algorithm for the trimming technique when m is even:

Input $y \in \mathbb{R}$ (either $\Re(y_j)$ or $\Im(y_j)$)

function TRIM(y, ϵ)

$\Delta = (2^{\frac{m}{2}} - 1)/2$

$r = y - \Delta$

$t = \Delta + \epsilon$

if $|r| > t$

$s = t/|r|$

$b = s \times r$

else

$b = r$

end if

return $b + \Delta$

end function

Using BWCD, we have obtained BER for dimensions when $m = 2, 4,$ and 6, and compared them with the BER of the SBWD. The plots as shown in Fig. 10 indicate that BWCD outperforms

SBWD by 0.5 dB. For the presented results, we have used $\epsilon = \frac{1}{2\sqrt{2}}$, which corresponds to the packing radius of $\frac{\sqrt{N}}{2}$. The above value of ϵ was optimized based on the simulation results by comparing the BER for various values of ϵ. Intuitively, trimming the received vector to fall within the packing radius of a lattice point in the edge of the lattice code forces to SBWD to decode to a lattice point in the edge of the code rather than a lattice point outside the lattice code.

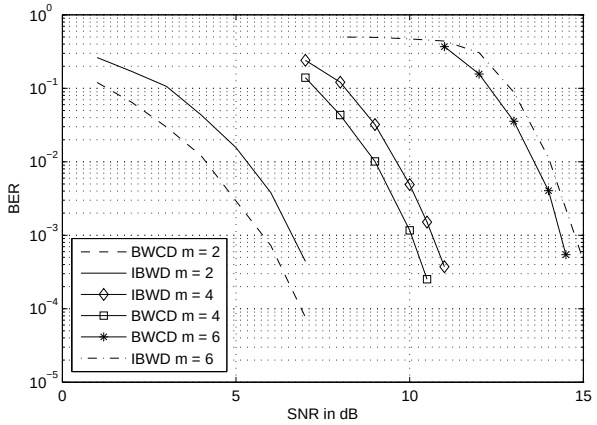

Fig. 10. Comparison of BER between BWCD and SBWD for $m = 2, 4,$ and 6.

VII. Conclusion and Directions for Future Work

In the first part of this paper, we have introduced a new method of encoding complex BW lattices, which facilitates bit labelling of BW lattice points. As a generalization, the proposed technique is applicable to encode all Construction D complex lattices. In the second part of this paper, we have used complex BW lattice codes for communication over AWGN channels. To encode the code, we have used Construction A', and to decode the code we have used the SBWD. We have studied the error performance of the SBWD, and have shown that the Jacobi-Theta functions can characterize the virtual binary channels that arise in the decoding process. We have also shown that the SBWD is powerful in making correct decisions beyond the packing

radius. Subsequently, we have used the SBWD to decode the complex lattice code through the noise trimming technique. This is the first work that uncovers the potential of SBWD (in terms of the error performance) in decoding lattice codes of large-block lengths with low-complexity. This work can be extended in one of the following ways:

- The SBWD proposed in [13] uses a soft-input, hard-output RM decoder at each level of Construction D. It will be interesting to study the error performance of the lattice decoder with soft-input, soft-output iterative RM decoders.
- We have presented the error performance of the SBWD through simulation results, and hence we now know the SBWD error performance with reference to the sphere lower bound and the sphere upper bound. A closed form expression on the error performance of the SBWD could be obtained for a better understanding of the decoder performance.

ACKNOWLEDGMENT

This work was performed within the Monash Software Defined Telecommunications Lab and supported by the Monash Professional Fellowship 2012-2013 and DP 130100103.

REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ: Wiley, 2006.

[2] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Transactions on Information Theory*, vol. 54, no. 4, Apr. 2008, pp. 1561-1585.

[3] M. Sadeghi, A. Banihashemi, and D. Panario, "Low-density parity-check lattices: construction and decoding analysis," *IEEE Trans, on Information Theory*, vol. 52, no. 10, Oct. 2006, pp. 4481-4495.

[4] O. Shalvi, N. Sommer, and M. Feder, "Signal codes: convolutional lattice codes," *IEEE Trans. on Information Theory*, vol. 57, no. 8, Aug. 2011, pp. 5203-5226.

[5] G. D. Forney, and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," *IEEE Trans. on Information Theory*, vol. 44, no. 6, Oct. 1998, pp. 2384-2413.

[6] E. S. Barnes, and G. E. Wall, "Some extreme forms defined in terms of Abelian groups," *J. Austral. Math. Soc.* 1, 1959, pp. 47–63.

[7] G. E. Wall, J. Pitman, R. B. Potts, "Eric Stephen Barnes 1924-2000," *Historical Records of Australian Science*, vol. 15, no. 1, June 2004, pp. 21–45.

[8] E. S. Barnes, and N. J. A Sloane, "New lattice packings of spheres," *Canadian Journal of Mathematics*, vol. 35, 1983, pp. 117–130.

[9] J. H. Conway and N.J.A Sloane, *Sphere Packings, Lattices and Groups*, Second Edition, 1993, Springer-Verlag, New York.

[10] G. D. Forney, "Coset Codes- Part II: Binary lattices and related codes," *IEEE Trans. on Information Theory*, vol. 34, no. 5, Sept. 1988, pp. 1152–1187.

[11] G. Nebe, E. M. Rains and N. J. A. Sloane, "A simple construction of the Barnes-Wall lattices," in *Codes, Graphs, and Systems: A Celebration of the Life and Career of G. David Forney, Jr. on the Occasion of his Sixtieth Birthday*, 2002, pp. 333–342.

[12] A. H. Banihashemi, and F. R. Kschischang, "Tanner graphs for group block codes and lattices: Construction and complexity," *IEEE Trans. on Information Theory* vol. 47, no. 02, 2001, pp. 822–834.

[13] D. Micciancio, and A. Nicolosi, "Efficient bounded distance decoders for Barnes-Wall lattices," in the Proc. of *IEEE ISIT 2008*, Toronto, Canada, July 6-11, 2008.

[14] A. J. Salomon, and O. Amrani, "Augmented product codes and lattices: Reed-Muller codes and Barnes-Wall lattices," *IEEE Trans. on Information Theory*, vol. 51, no. 11, Nov. 2005, pp. 3918–3930.

[15] E. Grigorescu and C. Peikert, "List decoding Barnes-Wall lattices," in the Proc. of *IEEE Conference on Computational Complexity 2012*, pp. 316–325, Porto, Portugal. Also available online at arXiv:1112.1994v1, Dec. 2011.

[16] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley Publishing Company, Inc, 1983.

[17] G. D. Forney, and A. Vardy, "Generalized minimum-distance decoding of Euclidean-space codes and lattices," *IEEE Trans. on Information Theory*, vol. 42, no. 6, Nov. 1996, pp. 1992–2026.

[18] G. D. Forney "A bounded-distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Trans. on Information Theory*, vol. 35, no. 4, July. 1989, pp. 906–909.

[19] M. Ran, and J. Snyders, "Efficient decoding of the Gosset, Coxeter-Todd and the Barnes-Wall lattices," in the Proc. of *IEEE ISIT 1998*, Cambridge, USA.

[20] M.D. Yucel "New decoding strategy for the 32-dimensional Barnes-Wall lattice," *IEEE Electronic letters*, vol. 29, no. 13, June. 1993, pp. 1231–1232.

[21] G. Schnabl, and M. Bossert, "Soft-decision decoding of RM Codes as genaralized multiple concatenated Codes," *IEEE Trans. on Information Theory*, vol. 41, no. 1, Jan. 1995, pp. 304–308.

[22] E. Viterbo, and E. Biglieri, "Computing the voronoi Cell of a lattice: the diamond-cutting algorithm," *IEEE Trans. on Information Theory*, vol. 42, no. 1, Jan. 1996, pp. 161–171.

[23] A.M Nielsen, and P. Kornerup "On Radix representations of rings," in the Proc. of *IEEE Symposium on Computer Arithmetic 1997*, pp. 34–43, Asilomar, CA, USA.

[24] J. Harshan, E. Viterbo, and J-C. Belfiore, “Construction of Barnes-Wall lattices from linear codes over rings,” in the Proc. of *IEEE ISIT-2012*, Cambridge, USA, July 2012

[25] J. Harshan, E. Viterbo, and J-C. Belfiore, ”Practical decoders for Barnes-Wall lattice constellations,” in *20th International Symposium on Mathematical Theory of Networks and Systems–2012*, Melbourne, Australia, July 2012.

[26] W. Reinhardt and P. Walker, *Theta Functions*, Cambridge University Press, 2010.